Journal of Engineering Science and Technology
Vol. 18, No. 6 (2023) 3113 - 3127
© School of Engineering, Taylor's University

OFFLINE KURDISH CHARACTER HANDWRITTEN RECOGNITION (OKCHR) USING CNN WITH VARIOUS PREPROCESSING TECHNIQUES

REEM M. IBRAHIM[1], GORAN SAMAN NARIMAN[2,*], HAMSA D. MAJEED[3]

[1]Department of Control & System Engineering of the University of Technology, Baghdad, Iraq
[2,3]Department of Information Technology, College of Science and Technology, University of Human Development, Sulaymaniyah, Kurdistan Region, Iraq
*Corresponding Author: goran.nariman@uhd.edu.iq, goransaman32@gmail.com

Abstract

Handwriting character recognition is an active and challenging area of research in the pattern-recognition field. Nowadays, many algorithms are introduced to achieve the highest accuracy. An improved recognition result could be accomplished if the input character images have good quality. That is why the pre-processing step becomes crucial for image identification missions. The algorithm of CNN (Convolutional Neural Networks) is fortified with numerous designs, permitting researchers to select the most operative architecture for better classification. However, this study suggests that the pre-processing mechanism is an important factor to be considered to increase the identification accuracy. This study utilised seven stages for pre-processing mechanisms applied to the dataset then the classification-accuracy measurement resultant from CNN was acquired. This work exposed that the utilisation of pre-processing steps indicated the most heightened accuracy with Binarized-KDS as an input, achieving training, testing, and validation accuracy of 99.2%, 97%, and 97.2% respectively after 35 iterations. Furthermore, another significant finding is that using different steps in processing input images to train the recognition model affects the recognition- rate within the same classifiers. Besides, the outcome reveals that each technique applied with the specific classifier may require a certain pre-process to obtain its optimal accuracy recognition rate.

Keywords: Convolutional neural network (CNN), Image processing, Kurdish handwritten recognition, optical character recognition (OCR), Pre-processing technique.

1. Introduction

One of the most researchable and problematic topics of Optical Character Recognition (OCR) is handwriting recognition [1]. OCR is a conversion system that converts non-editable machine data formats to editable formats, to further arithmetic and logical operations. Such a system has been used and applied to nearly all language scripts worldwide, among them, the English language is on top of the languages in which the aforementioned works have been applied on. Additionally, Kurdish is one of the languages with the least amount of research applications conducted. The lack of a solid dataset like EMNIST [2], which is prominent in data quantity, pre-processed, and prepared with various formats, is the primary cause of this study's limitation and high accuracy rate of Kurdish Handwriting Recognition. Regarding these issues, fortunately, recently an amazing work was published that contributed an excellent recognition rate and a huge Kurdish dataset collection has been conducted [3, 4].

In almost all languages, OCR systems have undergone extensive development; however, Kurdish is one of the languages in which the least has been done. The Kurdish language has two writing systems: Latin and Arabic alphabets. The Arabic scripts are used in Iraq and Iran, the Latin is used in Syria, Armenia, and Turkey [5, 6]. This research is restricted to Sorani (Central Kurdish) which is the standard used dialect for writing the Kurdish language. It is spoken by an estimated 9 to 10 million people in Iraq and Iran. Furthermore, Sorani is written in a cursive way, which implies that some characters may not have the same shapes depending on where they are positioned inside a word. However, this case has not been considered and only standalone letters have been collected in the selected dataset of [4] that used in this work.

Moreover, a recognition system model was built up by employing a Convolutional Neural Network (CNN), and a 96% accuracy rate was obtained in the testing results. For this, a collected dataset that consists of 35 classes of handwritten Kurdish characters with a total of 40,940 images has been used. The pre-processing phase of the dataset was running only to eliminate the table border of the collected character form using the Eraser Tool in Adobe Photoshop program and then cropping each letter cell to obtain 126 isolated images and letters from each form [4]. It may be observed that the pre-processing phase was not taken into consideration there, while there is a fact that the quality of pre-processing is proportional to the accuracy rate of the classifier.

The task of recognizing human handwriting was incredibly challenging; to overcome it, a reliable hand-tracking system that relies on strong features and classifiers is needed. Human handwriting recognition is an exceedingly tough research duty. A strong hand-tracking system based on an effective feature and classifier is required to solve such a problem. Nowadays, the most famous and achievable techniques are machine learning methods; Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Multilayer Perceptron (MLPs), Support Vector Machine algorithm (SVM), K-Nearest Neighbours algorithm (KNN), etc. [7]. Here, CNN is described shortly as employed in this work.

CNNs sometimes termed as ConvNets, are multi-layer neural networks used for object detection and image processing. Yann LeCun, the director of Facebook's AI Research Group, created the first CNN, known as LeNet [8], in 1998. LeNet was employed to perform character recognition tasks such as reading zip codes and

numbers. CNN contains multilayers that process, analyse, and extract characteristics from data by employing a convolutional layer that has multiple filters to conduct the convolution procedures. Furthermore, the Rectified Linear Unit (ReLU) layer is used to obtain a rectified feature map (RFM) from processing elements. The RFM is then fed into a pooling layer. Pooling is a down-sampling procedure that decreases the feature map's dimensionality. After that, it flattens the resultant two-dimensional arrays from the pooled feature map to create a long, single, continuous, and linear vector [9].

In addition, it is worth mentioning various architectures of CNN which have been proposed in recent years. In order to improve feature extraction, VGGNet (2014) from the Visual Geometry Group emphasized network depth with 3x3 convolutional filters [3]. In the same year, GoogleNet (Inception) introduced inception modules, using various filter sizes to effectively capture multiscale features [10]. By using skip connections to solve the vanishing gradient problem, Kaiming He et al.'s ResNet (2015) made it possible to train extremely deep networks [11]. Through densely connected layers, DenseNet (2016) maximized feature reuse while lowering parameter count [12]. MobileNet (2017) uses depth-wise separable convolutions and is optimized for devices with limited resources [13]. By systematically scaling model dimensions, EfficientNet (2019) achieved a trade-off between model size and performance [14]. Transformers, originally created for computer vision tasks, were modified to perform natural language processing in 2020 with models like ViT (Vision Transformer), demonstrating the adaptability of the architecture [15].

The contributions of this work are defined in three points; the first one defines the primary contribution and, in addition to it, two more points are extracted from it that are observable as described below:

- Implement CNN by proposing a sequence of image pre-processing to achieve a high accuracy rate.
- Approving the fact that the quality of pre-processed images is proportional to the accuracy rate of the classifier.
- Each feature extraction and classifier technique may require a unique pre-process to obtain its optimal accuracy recognition rate.

2. Related Works

In the field of handwritten recognition, many approaches have been published to accomplish the aim of high recognition accuracy. Yet for the Kurdish alphabet, a lack of proposals appears which will be covered in this section. Ahmed et al. [3] proposed a model to recognize an offline isolated handwritten Kurdish character using the Deep CNN model through a self-constructed dataset. The model used compilation of sparse cross-entropy as a loss function and ADAM optimizer along with increasing the epochs of training which show an effect on the accuracy of the model. The approach achieved 96% accuracy. Mahmood et al. [16] presented a system for Kurdish Sign Language using two lines of features that extracted from two lines of real-time video to detect dynamic hand movements. A pre-processing step for the hand object was presented through the employment of binary-captured frame values optimization. an Artificial Neural Network utilised as a classifier with 98% accuracy. Meanwhile [17] proposed an online recognition system for Kurdish characters. The proposed approach seized a hybrid of a hidden Markov model along

with harmony search algorithms. The first part of the classification process is achieved by the Markov model based upon a common directional feature vector, this step reduces the time of the upcoming recognition phase which is achieved by the harmony search recognizer using a dominant with common pattern movement by means of a fitness function. The recognition rate of 93.52% was accomplished by the proposed system with an average of 500 ms. Another study addresses the challenge of recognizing the complex and unique Sundanese script, vital for cultural preservation [18]. Using CNN and data augmentation techniques like flipping, rotation, and translation, the research significantly improves accuracy by approximately 17.07%, reaching the best accuracy of 80% with a baseline model. This approach proves effective in enhancing the recognition of Sundanese handwriting patterns, contributing to the preservation of this cultural heritage.

Meanwhile, authors explore the field of pre-processing in order to develop existing models. As for the recognition field of research, the higher accuracy of recognition is the dominant demand, and adding the pre-processing stage becomes a major contribution to satisfy this demand. A lot of research has been done in a lot of work for this pre-treatment. Kusrini et al. [19] suggested a study to improve the classification accuracy by utilising the Gaussian filter for the pre-processing mechanism plus VGG16 as a learning architecture in CNN networks. A combination of Gaussian filters with different pre-processing mechanisms was applied to the dataset, and accuracy measurement resultant from the VGG16 learning architecture utilisation was attained. The model succeeded in achieving the highest accuracy with 98.75%. Meanwhile, [20] attempted to summarise various aspects and procedures of pre-processing implemented in many recognition techniques. The purpose of their work was to provide help to the researchers in gaining knowledge of dissimilar pre-processing techniques that could be adopted in offline OCR.

It is worth mentioning here some relevant thoughts from the literature; the authors of [5] showed different techniques for pre-processing phase of OCR systems and mentioned that "OCR system can be made robust through applying correct Image enhancement techniques". Furthermore, the literature [6] emphasizes that "It is also probable that more powerful classifiers profit from the pre-processing". In addition, undoubtedly, for the chosen feature extraction methods, various pre-processing techniques are required to be tested to determine the most proper one to get the optimal accuracy rate for the selected model.

3. The Proposed Method

The proposal of this study is divided into two main divisions: the pre-processing phase and the classification phase. There are several classifications of handwritten recognition with different attributes to be selected as [21] suggested; adding the classification table with the selected option in this domain of work. The abstract view of this work is clarified through Table 1 which shows the used category of the classes for the presented method.

Table 1 illustrates that the Script Writing System defines the selected language to be used in the proposed model in our case the language is Kurdish (Sorani). The way of getting data is defined as Data Acquisition, which can be either offline or online. Offline means the input data is a form of images, while online means the input data is through the touch screen and is recognized instantly. In the current study, the offline mode has been selected. The Granularity Level of Documents

defines the level of detailed data to be processed, for example, a paragraph, sentence line, word, or a single character. A single character is used for the proposed model to be processed. There are two types of data sources; public and self-constructed datasets. This has been classified under the name of Source of The Collected Dataset. In this work, a public dataset has been chosen. Finally, the Script Recognition Process refers to the phases that have been implemented to achieve the recognition. As mentioned in [22], the whole possible implementable process consists of four sequential phases, vis.: Pre-processing (P), Segmentation (S), Feature Extraction (F), and Classification (C). Due to the character-based Granularity Level being applied, the segmentation phase is not required, consequently, the Script Recognition Process is depicted as Pre-processing- Feature extraction- Classification (PFC).

Table 1. Classification of the proposed technique.

Classifications	Nominated Category
Script Writing System	Kurdish Alphabet
Data Acquisition	Offline-Handwritten
Granularity Level of Documents	Character Level
Source of The Collected Dataset	Public Dataset
Script Recognition Process	PFC

The proposed architecture is represented in Fig. 1, the script recognition process of the proposed approach produced in Fig. 1(a) which starts from the input of the handwritten Kurdish Alphabet images. A public dataset presented by [4] has been used, for simplicity, we abbreviate it to (KDS) which stands for Kurdish Dataset. The preceding section (Introduction) highlights the property of the dataset. The next step is pre-processing, which is designed to generate two different pre-processed formats vis. Skeleton-KDS and Binarized-KDS. Both are fed to the CNN model to proceed with the recognition.

3.1. The proposed pre-processing model.

Pre-processing approved by many studies that the model's accuracy rate directly leans on the output quality of this phase. This work is trying to add another approval to this fact through this proposed phase which consists of seven stages as Fig. 1(b) shows. Those seven stages of pre-processing are applied in an attempt to improve the quality of the input image then as a result the recognition-rate is improved. The pre-processing steps are presented in Fig. 2 and applied as follows:

3.1.1. Greyscale conversion

Digital photography technique applied to KDS images where all colour statistics are removed, only numerous grey- shades maintain, the black presence the darkest while white is the lightest. Meanwhile, the brightness of the intermediate shades is equivalent to the primary colours (red, blue, and green). Consecutively, the technique employs an equivalent quantity of the essential pigments, yellow, magenta, and cyan. The image brightness is signified by each pixel.

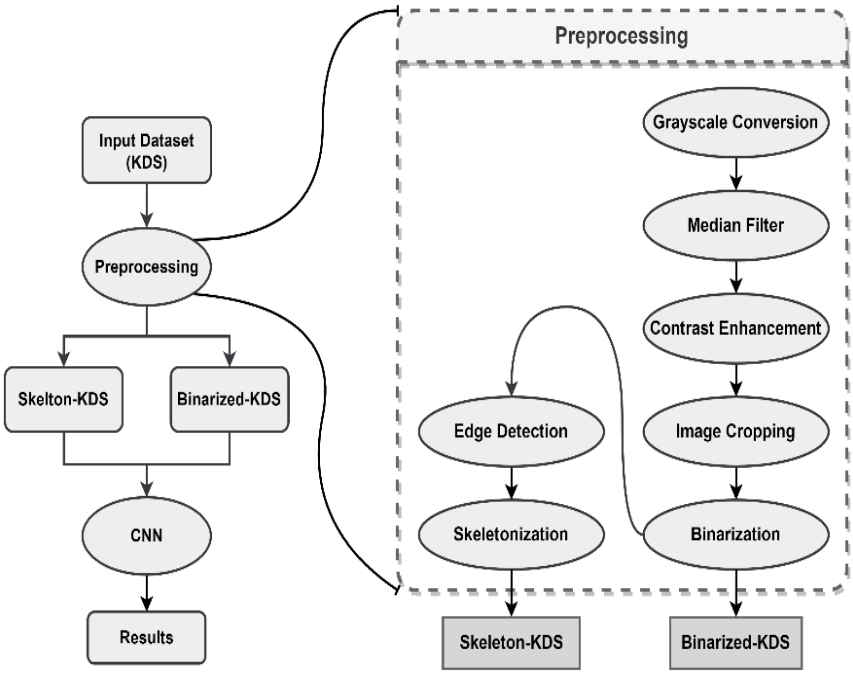


Fig. 1. The proposed model (a) Script recognition process of the proposed approach and (b) The pre-processing phases.

3.1.2. Median filter

A method of non-linear digital filter is a noise reduction procedure used to eliminate noise from KDS images and to enhance the outcomes of subsequent processing. The ability to preserve edges along with reducing noise makes median filtering incredibly widespread in digital image processing.

3.1.3. Contrast enhancement

Increasing the difference of brightness between objects and their circumstances, contrast enhancements increase the visibility of objects in the KDS image. Although these might both be done in one step, contrast enhancements are commonly conducted as a contrast stretch followed by a tonal enhancement. While tonal enhancements boost the brightness differences in the dark (shadow), mid-tone (grey), or bright (highlight) areas at the outflow of the brightness differences of the other regions, a contrast stretch advances the brightness changes equally crossways the image`s dynamic range. Histogram Equalization technique has been employed in this work that is used to equalize the brightness level

3.1.4. Image cropping

KDS digital picture cropping removes the image's outer edges. Cropping could be employed to decrease the size (in pixels) and/or change the aspect ratio of an image (ratio of length to width). In this work, both of them are applied, by cropping the

area of interest the image`s size is reduced (in pixels), and the aspect ratio is adjusted to 1:1 while preserving the image resolution.

3.1.5. Binarization

KDS image is converted from a multi-tone image into a two-tone image using the binarization method. The process is performed based on a global-single threshold with a mean value equal to 220, The threshold determines if a pixel's grey value is larger than the threshold value, in which case the pixel is turned white. Similarly, if a pixel's grey value is lower than the threshold, that pixel gets turned black. Additionally, another step applied to the aim of reconstruction of the broken pixels if any.

In this work, the output of this step (Binarized-KDS) is fed to CNN to get our first recognition-rate result. Another rate will be obtained from the output of extra steps of pre-processing. This is to show the effect of different pre-processing technique applications on the same classifier from one end and on different classifiers from the other end.

3.1.6. Edge detection

The edge detection technique is used to recognize points in a digital image of KDS with disjointedness, The brightness of the image shifts abruptly. The edges (or boundaries) of the image represent the sections where the image`s brightness fluctuates sharply. The canny technique is employed in this work to perform the edge detection task, in comparison to many other approaches, this one is the most popular and very effective. It is a multi-stage method that can find/detect many different kinds of edges. This step is applied as a preparation for the next and last step of this phase.

3.1.7. Skeletonization

When foreground portions in a binary KDS image are shrunk down to skeletal remnants, the original region's extent and connectivity are substantially preserved while the majority of the foreground pixels are discarded. This process is known as skeletonization and Morphological operations on binary images using ‘skel’ operation to get the image skeleton have been applied in this study. The output of this step (Skeleton-KDS) is considered as the second outcome from the pre-processing phase that is fed to CNN to be compared later with the first output (Binarized-KDS) as an attempt of this work to demonstrate the impact of applying various pre-processing techniques to the same classifier.

3.2. 3. KDS recognition using CNN

For the recognition phase, the choice must be CNN, all the hyperparameters settings were identical to [3], and the employed CNN is structured by implementing the following layers:

- Convolutional Layers: It is used to share weights (kernels or filters) to create feature maps throughout the input data. This weight-sharing reduces the number of parameters while improving network performance. The filters can detect features like colours, edges, or patterns.

- Non-linear Activation Layer: Convolutional layers are linear filters, so non-linear activation functions (e.g., sigmoid or ReLU) are used to introduce non-linearity into the model. Sigmoid functions suffer from saturation problems, whereas ReLU is a popular alternative.
- Pooling Layers: Pooling layers are employed to reduce dimensionality, increase computational efficiency, and make the network less sensitive to input variations. They aggregate information from previous layers, preserving essential knowledge.
- Flatten Layers: After the convolutional, activation, and pooling layers, a flattening step converts the multidimensional feature maps into one-dimensional vectors, ready for the fully connected layers.
- Output Layer: The fully connected layers at the top of the CNN generate a probability distribution over output classes using the softmax function. This allows for classification based on the features extracted by earlier layers.
- Dropout: It is a regularization technique to mitigate overfitting in deep neural networks. It is used to randomly drop units and their connections during training, preventing the co-adaptation of neurons.

Furthermore, the specifications for the CNN parameters were set as follows:

- The CNN consists of three convolutional blocks, each comprising a Conv2D layer and a max-pooling layer.
- The first convolutional block has 16 filters, the second one has 32 filters, and the third one has 64 filters.
- All convolutional filters have a size of 3 x 3.
- All max-pooling layers use a pooling size of (2, 2).
- After the three convolutional blocks, there is a flattened layer followed by a fully connected layer with 512 units.
- The CNN's final output provides class probabilities for five classes using the softmax activation function.
- All layers, except the output layer, use the ReLU activation function.
- A dropout layer with a 20% dropout probability is incorporated where appropriate.
- The CNN model is compiled using the ADAM optimizer, and the sparse cross-entropy loss function is used.

The utilized dataset is the same (KDS), and the CNN is implemented identically to observe the effect of pre-processing techniques on the recognition accuracy rate as the aim of this work is to improve the accuracy of the research by improving the pre-processing phase.

4. Results and Discussion

The proposed model has been implemented in two phases; the pre-processing phase followed by feature extraction along with classification using CNN.

The first group of findings was in the form of images, and they came from the pre-processing phase, which involves seven steps as Fig. 1(b) illustrates, starting with greyscale conversion and ending with skeletonization. The results of this phase are shown in Fig. 2 for the letter S (س).

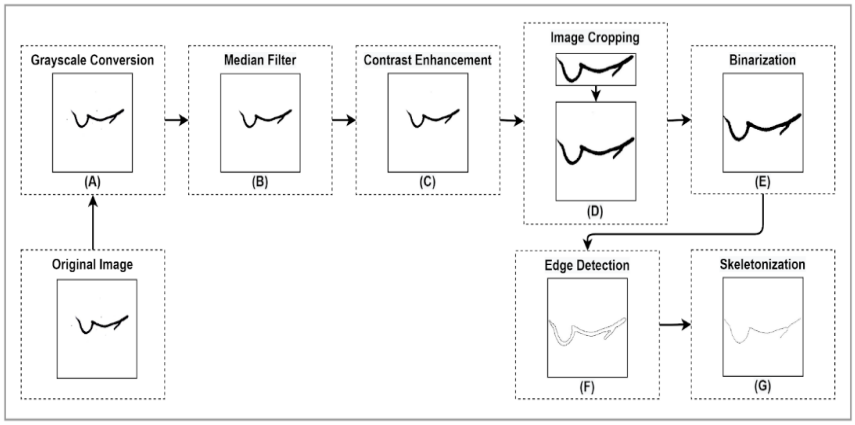


Fig. 2. A sample of KDS character through the pre-processing phases.

In the early stages of the pre-processing phase, the entered image will be transformed to greyscale regardless of its original colour; in the second stage, the noise is eliminated and the brightness level is equalized producing the image's improvement of contrast. Size refinement is used to determine the region of interest besides adjusting the image aspect ratio to 1:1 with maintaining the image`s resolution as a preliminary step to the next stage. The adaptive thresholding of the input demonstrates the results of binary conversion. For the used dataset a threshold with a mean value equal to 220 is applied. The output of this stage is considered the first set of the output of this phase to be fed to CNN named (Binarized-KDS) as shown in Fig. 2(E). After binarization, the edge of the interested region is discovered. The last stage represents the output of the resultant skeletonization which is deemed as a second set of output named (Skeleton-KDS), shown in Fig. 2(G).

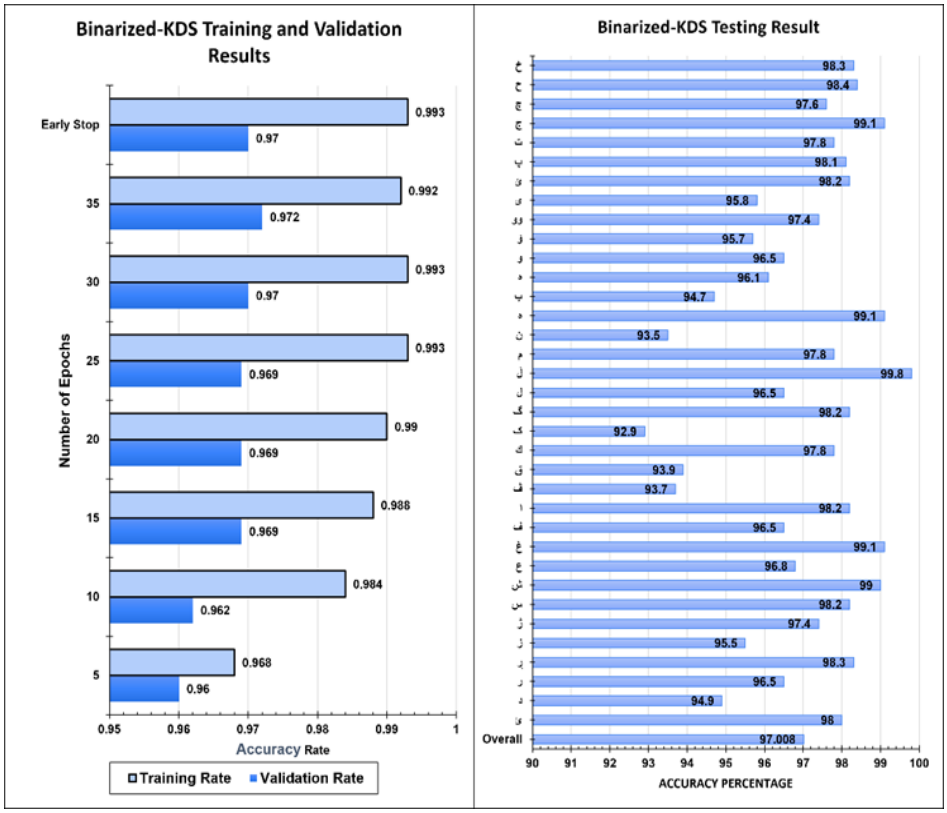
The aim behind applying two sets of outputs to be fed to the classifier is to show two points of view: The first one is to show that the different levels of pre-processing affect the resultant accuracy of the same classifier, while, the specific level of pre-processing considered as an optimum level to be fed to that specific classifier to get the finest accuracy-rate as a second. The 35 letters for each person are applied in this phase using the same stages. In actuality, it applied to the CNN state for all acquired datasets. The second group of outcomes appeared as numbers, where the Binarized-KDS and Skeleton-KDS are provided to the CNN and the phase of classification starts with revealing the percentage of recognition-rate. The classifier`s results will be presented and discussed in the upcoming subsection.

4.1. Result of binarized-KDS

The CNN`s results are presented in this subsection where Binarized-KDS is employed as an input to the recognition phase. As hypothesized, the performance of the CNN model using the properly pre-processed dataset has affected that of the poorly pre-processed dataset. The achieved training, testing, and validation accuracy are 99.2%, 97%, and 97.2% respectively after only 35 iterations.

The outcomes of this approach are presented in Fig. 3. The training and validation accuracy rates are shown in Fig. 3(a) for all running epochs up until the

early stopping condition occurs. Attempted to run 100 epochs with 5 epochs as a patient, which means, running 5 more epochs even if the performance of the model does not improve before getting to the Early Stop condition. In epoch 35 got the highest performance. To obtain the testing accuracy of the model, a confusion matrix was generated and summarised in Fig. 3(b). It shows the individual accuracy rate of each Kurdish character with their total average.



(a) **(b)**

Fig. 3. Binarized-KDS results in (a) Training and Validation accuracy rate and (b) The overall and individual Kurdish character testing accuracy rate.

4.2. Result of skeleton-KDS

In this section, the accuracy value for using Skeleton-KDS will be thoroughly explained using the visual representations in Fig. 4. The accuracy results for each of the experimental cases are presented. The experimental results for the achieved training and validation accuracy percentage, are 99.3% and 95.5%, respectively, before the early stop condition occurs, as shown in Fig. 4(a). Furthermore, to determine the testing accuracy for the Skeleton-KDS, the corresponding accuracy rates for each of the Kurdish letters are independently calculated as depicted in the graph of Fig. 4(b), and 95% is the average testing accuracy for all the letters in the dataset.

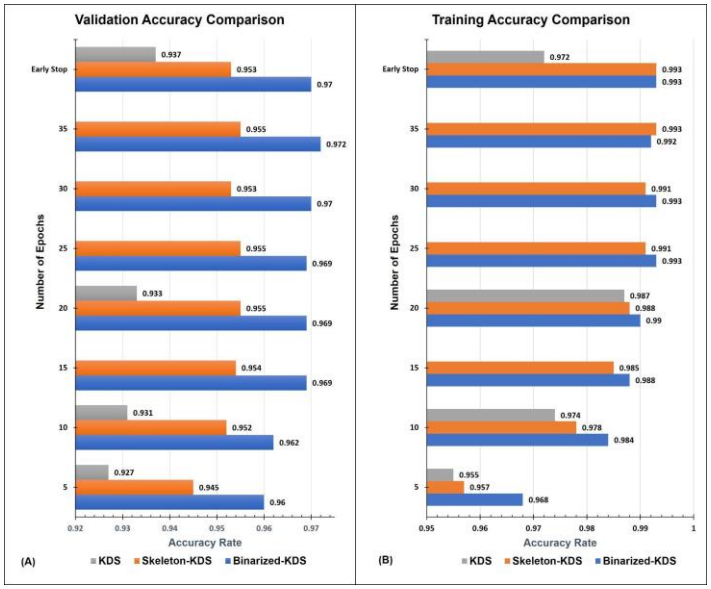
(a) **(b)**

Fig. 4. Skeleton-KDS results. (a) Training and Validation accuracy rate and (b) The overall and individual Kurdish character testing accuracy rate.

4.3. Discussion

The model`s output in general reveals that the use of Binarized-KDS achieved the highest accuracy percentage for all training, testing, and validation. Furthermore, the use of Skeleton-KDS obtained a higher testing and validation accuracy rate than the results of KDS obtained from [4]. In other words, KDS got a higher accuracy rate only in the testing experimental over Skeleton-KDS. Simply, we can claim that the best selection for the given CNN model is Binarization-KDS, Skeleton-KDS, and then KDS in that order. A comparison has been made and represented in Fig. 5 and Table 2.

Table 2, which is a summary of the charts in Figs. 5 and 6, shows that using Binarized-KDS continued to improve its performance until epoch 35, whereas Skeleton-KDS and the original KDS stopped improvements after only 20 epochs. This means that the features in the last two datasets were learnable by CNN for only 20 iterations, in other words, all extractable features by CNN in these datasets were enough for a total of 20 iterations. However, the hidden features of the Binarized-KDS could be extracted to improve the learning of the CNN model along with 35 epochs. This is a sign of a high-quality and properly pre-processed dataset for the given CNN model.



(a) **(b)**

Fig. 5. Comparison results (a) KDS, Skeleton-KDS, and Binarized-KDS validation accuracy comparison and (b) KDS, Skeleton-KDS, and Binarized-KDS training accuracy comparison.

Table 2. Accuracy rates for three separate pre-processed datasets during training and validation tests.

Number of Epochs	Validation Accuracy Rate			Training Accuracy Rate		
	Binarized-KDS	Skeleton-KDS	KDS	Binarized-KDS	Skeleton-KDS	KDS
5	0.96	0.945	0.927	0.968	0.957	0.955
10	0.962	0.952	0.931	0.984	0.978	0.974
15	0.969	0.954	N/A	0.988	0.985	N/A
20	0.969	0.955	0.933	0.99	0.988	0.987
25	0.969	0.955	N/A	0.993	0.991	N/A
30	0.97	0.953	N/A	0.993	0.991	N/A
35	0.972	0.955	N/A	0.992	0.993	N/A
Early Stop	0.97	0.953	0.937	0.993	0.993	0.972

We can conclude that based on our experiments, the accuracy of the CNN has been increased by properly pre-processing the dataset. Also, the use of the steps of pre-processing until getting binarized output, as detailed in Subsection 2.1, is the best choice for CNN.

Finally, we can state the following facts as claimed in Section 1:

- By proposing a sequence of image pre-processing and generating Binarized-KDS, a high accuracy rate has been achieved.
- Approved the fact that the quality of pre-processed images is proportional to the accuracy rate of the classifier. The quality of Binarized-KDS is better compared to poorly-pre-processed KDS.

- Each feature extraction and classifier technique may require a unique pre-process to obtain its optimal accuracy recognition rate. In our case, the Binarized-KDS is the optimal one.

Observing the overall comparison, as Fig. 6 shows, for the total accuracy percentage calculated from the three tested datasets, KDS from [4], Binarized-KDS, and Skeleton-KDS from this work for all the 35 Kurdish letters. Noticeably, Binarized-KDS gives higher accuracy among them. As for KDS and Skeleton-KDS, the percentages were differentiated among the characters and that was because the similarity between some of the letters led to the misclassification.

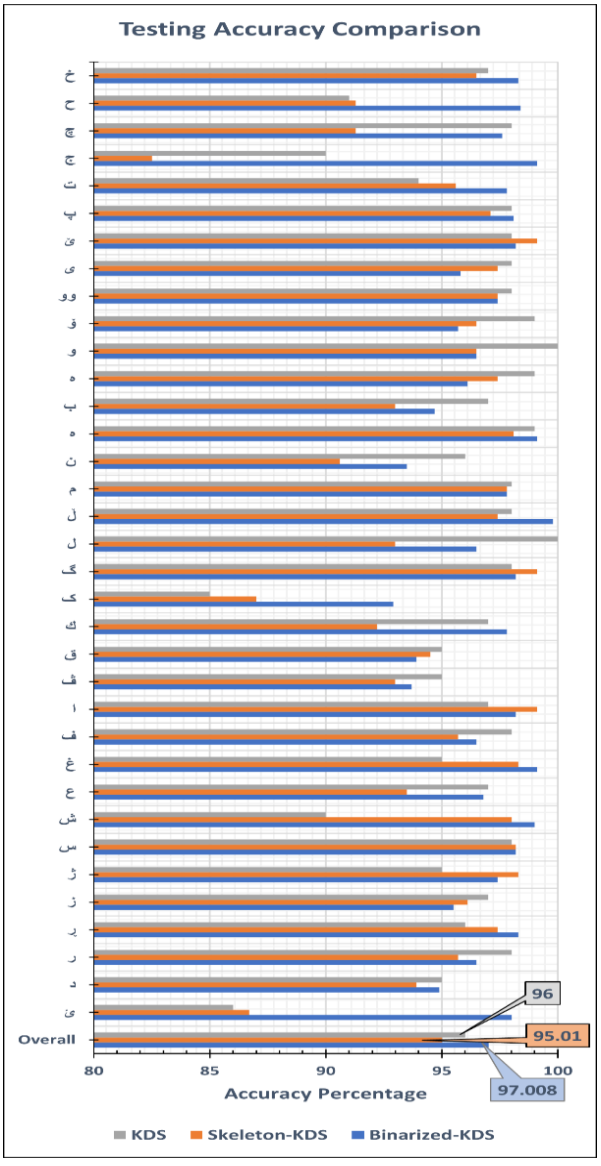


Fig. 5. Comparison results of testing accuracy for KDS, Skeleton-KDS, and Binarized-KDS.

5. Conclusion

To evaluate the effectiveness of the suggested model, experimental results have been established. The model is implemented using a pre-processing phase, followed by an evaluation step on a public dataset made up of 40,940 photos totalling 35 classes of handwritten Kurdish characters that were collected from individuals. This study is an attempt to show that applying the pre-processing phase on the input affects the output recognition-rate. The outcomes of this work verified that assumption by achieving an improvement in the output classification accuracy after using three sets of input datasets, not pre-processed KDS, and two different levels of pre-processed named, Binarized-KDS and Skeleton-KDS, adding the pre-processed give the higher accuracy rate with Binarized-KDS input achieving training, testing, and validation accuracy 99.2%, 97%, and 97.2% respectively after only 35 iterations. Furthermore, using different techniques of pre-processing on the input images affects the recognition-rate within the same classifiers. Another point achieved is that each technique applied with the specific classifier may require a certain pre-process step to obtain its optimal accuracy recognition rate. Despite the high identification rate, the experimental results show some misclassification of particular letters. These problems could be improved by making modest adjustments to the pre-processing methods currently being employed.

References

1. Tappert, C. C., Suen, C. Y., and Wakahara, T. (1990). The state of the art in online handwriting recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(8), 787-808.
2. Cohen, G., Afshar, S., Tapson, J., and van Schaik, A. (2017). EMNIST: an extension of MNIST to handwritten letters. Retrieved from http://arxiv.org/abs/1702.05373
3. Ahmed, R. M., Rashid, T. A., Fattah, P., Alsadoon, A., Bacanin, N., Mirjalili, S., Vimal, S., and Chhabra, A. (2022). Kurdish Handwritten character recognition using deep learning techniques. Gene Expression Patterns: GEP, 46(119278), 119278.
4. Ahmed, R. M., Rashid, T. A., Fatah, P., Alsadoon, A., and Mirjalili, S. (2021). An extensive dataset of handwritten central Kurdish isolated characters. Data in Brief, 39(107479), 107479.
5. Esmaili, K. S., Eliassi, D., Salavati, S., Aliabadi, P., Mohammadi, A., Yosefi, S., and Hakimi, S. (2013). Building a Test Collection for Sorani Kurdish. 2013 ACS International Conference on Computer Systems and Applications (AICCSA). IEEE.
6. Hassani, H., and Medjedovic, D. (2016). Automatic Kurdish dialects identification. Computer Science and Information Technology (CS and IT). Academy and Industry Research Collaboration Center (AIRCC).
7. Lecun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
8. LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324

9. Brownlee. J. (2018). Better Deep Learning: Train Faster, Reduce Overfitting, and Make Better Predictions. Machine Learning Mastery.
10. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., and Anguelov, D. (2015). Going deeper with convolutions. arXiv preprint arXiv:1409.4842.
11. He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385.
12. Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. arXiv preprint arXiv:1608.06993.
13. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
14. Tan, M., and Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946.
15. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. In arXiv [cs.CV]. http://arxiv.org/abs/2010.11929
16. Mahmood, M. R., Mohsin Abdulazeez, A., and Orman, Z. (2018). Dynamic hand gesture recognition system for Kurdish sign language using two lines of features. 2018 International Conference on Advanced Science and Engineering (ICOASE). IEEE.
17. Zarro, R. D., and Anwer, M. A. (2017). Recognition-based online Kurdish character recognition using hidden Markov model and harmony search. Engineering Science and Technology an International Journal, 20(2), 783-794.
18. Maliki, I., & Prayoga, A. S. (2023). Implementation of Convolutional Neural Network for Sundanese Script Handwriting Recognition with Data Augmentation. Journal of Engineering Science and Technology, 18(2), 1113-1123.
19. Kusrini, K., Arif Yudianto, M. R., and Al Fatta, H. (2022). The effect of Gaussian filter and data preprocessing on the classification of Punakawan puppet images with the convolutional neural network algorithm. International Journal of Electrical and Computer Engineering (IJECE), 12(4), 3752.
20. R. Dey, R. C. Balabantaray, S. Mohanty, D. Singh, M. Karuppiah, and D. Samanta. (2022). Approach for preprocessing in offline optical character recognition (OCR). Interdisciplinary Research in Technology and Management (IRTM).
21. D. Majeed, H., and Saman Nariman, G. (2022). Offline handwritten English alphabet recognition (OHEAR). UHD Journal of Science and Technology, 6(2), 29-38.
22. D. Majeed, H., and Saman Nariman, G. (2023). Construction of alphabetic character recognition systems: A review. UHD Journal of Science and Technology, 7(1), 32-42.